Hindawi
Security and Communication Networks
Volume 2021, Article ID 9918697, 11 pages
https://doi.org/10.1155/2021/9918697

Research Article

Implementation of Blockchain Consensus Algorithm on Embedded Architecture

Tarek Frikha,[1] Faten Chaabane,[2] Nadhir Aouinti,[1] Omar Cheikhrouhou,[3] Nader Ben Amor,[1] and Abdelfateh Kerrouche[4]

[1] *Université de Sfax, CES Lab, 3038, Sfax, Tunisia*
[2] *Université de Sfax, Regim Lab, Sfax, 3038, Tunisia*
[3] *College of CIT, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[4] *C82c School of Engineering and the Built Environment Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT, UK*

Correspondence should be addressed to Tarek Frikha; tarek.frikha@enis.tn

Received 6 March 2021; Revised 29 March 2021; Accepted 8 April 2021; Published 23 April 2021

Academic Editor: Azeem Irshad

Copyright © 2021 Tarek Frikha et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The adoption of Internet of Things (IoT) technology across many applications, such as autonomous systems, communication, and healthcare, is driving the market's growth at a positive rate. The emergence of advanced data analytics techniques such as blockchain for connected IoT devices has the potential to reduce the cost and increase in cloud platform adoption. Blockchain is a key technology for real-time IoT applications providing trust in distributed robotic systems running on embedded hardware without the need for certification authorities. There are many challenges in blockchain IoT applications such as the power consumption and the execution time. These specific constraints have to be carefully considered besides other constraints such as number of nodes and data security. In this paper, a novel approach is discussed based on hybrid HW/SW architecture and designed for Proof of Work (PoW) consensus which is the most used consensus mechanism in blockchain. The proposed architecture is validated using the Ethereum blockchain with the Keccak 256 and the field-programmable gate array (FPGA) ZedBoard development kit. This implementation shows improvement in execution time of 338% and minimizing power consumption of 255% compared to the use of Nvidia Maxwell GPUs.

1. Introduction

The global IoT market is expected to reach a value of USD 1,386.06 billion by 2026 from USD 761.4 billion in 2020 at a CAGR of 10.53%, during the period 2021–2026 [1].

The IoT technology is connecting various devices such as mobile phones, sensors, and household appliances together for collecting and sharing data for the next industrial revolution of intelligent connectivity. The fourth industrial revolution (Industry 4.0) interconnects smart digital technology with real worlds to create smart manufacturing and supply chain management [1]. In the current context, the emergence of Industry 4.0 and the adoption of IoT devices require manufacturers to implement innovative ways to advance production with intelligent connectivity that uses more robotics and avoids industrial accidents and machines' downtime failure. Therefore, industries, hospitals, supply chains, governments, banks, and logistics need to be connected using Distributed Ledger Technologies (DLT) such as blockchain technology to react quickly for a more connected world. This will enable more secured process dealing with big data analysis generated by IoT devices.

Blockchain is mainly dealing with data storage and management and a distribution technology that is transparent and secure and operates regardless of a central control body [2].

Unlike traditional methods, blockchain allows peer-to-peer transfer of digital assets without the need for an intermediary. This technology was inspired by Bitcoin [3] cryptography and then has emerged, evolved, and spread in several applications including finance [4], health [5], administration [6], industry [7], agriculture [8], and smart cities [9]. It affects also other sectors such as the transfer of goods (supply chain), digital media transfer (sale of works of art), remote service delivery (travel and tourism), distributed intelligence (graduation), electricity generation and distribution, startup fundraising, electronic voting, identity management, crowdfunding (increasing startup funds), and crowd-operation (remote voting).

The first blockchain success notified with Bitcoin, was followed by other blockchains such as Ethereum [10], Hyperledger Fabric [11], Azur, Grid+, IOTA [12], and Tezos [13, 14].

Representing the new generation of blockchain, Ethereum can play a major role of a public blockchain like Bitcoin, or a private blockchain such as Hyperledger Fabric. It is also the basis of other blockchains which are specific frameworks for applications, such as the Azur. For example, the blockchain proposed by Microsoft, which was optimized to take advantage of the characteristics of the cloud. Another example is the Grid + blockchain which is used in energy management applications.

To preserve the security of the blockchain, a specific algorithm, known as consensus, is used. It allows a new block to be added to the blockchain without compromising the integrity of data stored in the distributed ledger.

Moreover, some blockchains are defined with intelligent contracts and software platforms to play the role of links in the blockchain. However, all these blockchains are using consensus to preserve their security. In this context, several types of consensus are proposed in the literature such as the Proof of Work (PoW), the Proof of Stake (PoS), the Proof of Authority (PoA), the PBFT, and the Ripple and the Raft [14]. These consensus algorithms have different complexity levels. One of the most complex and energy-intensive consensuses is PoW which was used in several blockchains such as Bitcoin, Ethereum, and IoTA [15]. As an example, the mining process time is approximately 10 minutes for Bitcoin [16] and 15 seconds for Ethereum using Nvidia RTX 3080 GPU [17]. Regardless of the number of miners, it still takes about 10 minutes to mine one Bitcoin. At 600 seconds (10 minutes), all else being equal it will take 72,000 GW (or 72 terawatts) of power to mine a Bitcoin using the average power usage provided by ASIC miners [16].

The use of blockchain, particularly the mining part, requires significant computing resources. In this paper, a feasibility study of implementing the blockchain on an embedded system and particularly on field-programmable gate array FPGA is presented taking into consideration all the resource requirements to validate this approach.

An embedded architecture is proposed to implement the PoW consensus, especially on FPGA-based architecture. This optimized architecture should accelerate the classical PoW process and consequently minimize the energy consumption. This proposed architecture is chosen according to a comparison between different software (SW), hardware (HW), and mixed architectures.

More precisely, the contribution of this paper is as follows.

The main contribution of this paper consists of two parts. First, an embedded architecture is proposed to implement the PoW consensus algorithm on FPGA. This part is called the off-chain system block. And, the second part is dedicated to the design of an off-chain/on-chain system. The PoW implementation and particularly the hash algorithm were off-chain (on FPGA). The node smart contract, transactions, and blocks where on-chain (they are implemented on the Raspberry Pi 3 platform).

The remainder of this paper is as follows. In Section 2, we describe the basic notions of the blockchain, particularly its different consensus followed by a study on embedded technologies and mixed HW/SW architectures [18]. In Section 3, a description of the PoW used in the blockchain Ethereum will be dissected. The profiling of this function will allow to describe the embedded architecture to be chosen. Section 4 is reserved for the choice of the architecture and the different parts of our system containing the consensus implementation. The last part will be reserved for the results obtained and the comparison between SW on GPU and HW-implemented architecture from execution time and energy consumption point of view. Finally, in Section 5, we conclude and give potential perspectives.

2. Background

2.1. Blockchain Overview. In this section, we give an overview of the blockchain technology and its different classes and main components.

2.2. Security-Based Blockchain Classification. From the security point of view, blockchain can be classified as public, consortium, and private.

2.3. Public Blockchain. The blockchain is said to be public because it is open to everyone. Thus, it is assimilated to a marketplace, where anyone can open a store to offer any products and services. In this case, there are no restrictions on the comings and goings of visitors who are free to visit the different stores to make purchases.

Consequently, a public blockchain has several characteristics, such as a decentralized network which is open to all actors without any restriction, data can be consulted by all without any restriction, and data can be consulted by all without any restriction, but it is indelible, forgery-proof and cannot be modified afterwards. In this class of blockchain, the use of the PoW consensus makes the blockchain's transactions impossible to falsify and very easy to manipulate.

There are many examples of public blockchains: Bitcoin, Ethereum, Ripple [14], Litecoin [19], and Dash.

2.4. Consortium Blockchain. It consists of a permitted blockchain which is partially decentralized and differs from public blockchains because its network is only accessible to a limited number of users.

New members must be validated by the nodes and already existing members in the consortium, and the accessibility of the data depends on the access rights granted to each node. It can be compared to a corporate marketplace (here, the "consortium") for which only consortium members would be allowed to open a store to offer products and services. However, the consortium may grant some exemptions to open additional stores. The comings and goings in this marketplace are normally restricted by the rules defined by the consortium.

It should be noted that the vast majority of existing consortium blockchains operate under the Proof of Authority (PoA) system. As examples of public blockchains, we can cite Ripple [20], Funds DLT, etc.

2.5. Private Blockchain. In contrast to public blockchains, private blockchains (of which permitted blockchains are a special case) are like distributed databases.

Their characteristics are as follows:

(i) The network is accessible to a limited number of users. New entrants must be validated by a central decision-making body.

(ii) The accessibility of the data depends on the access rights of each node. This is defined by the central decision-making body.

(iii) On a private blockchain, the consensus is based on the trust placed in all the validator nodes.

A private blockchain can be compared to a marketplace where all members authorized to launch a store, or to sell products and services, are only members of this same structure.

As a result, the cases of use are very frequent. As for distributed databases, they are useful for sharing confidential or important data within an organization or within the different entities of a group.

There are many examples of private blockchains. We can cite Hyperledger Fabric, Grid+, Azur, Ethereum (both private and public blockchains), etc.

2.6. Consensus Algorithms. It consists of the transition from centralized systems where the administrator or the central system can validate or invalidate transactions such as the banking system and database management systems.

In this kind of systems, the administrator is the valid or invalid manager. In decentralized systems such as blockchains, the absence of an administrator requires another protocol for verification and validation. The intermediary functions are moved to the periphery participating pair in the infrastructure of the chain. Since the peers do not necessarily know each other, it is a decentralized system.

The consensus algorithm consists of firstly setting up a process to validate, verify, and confirm transactions, then recording the transactions in a large distributed directory, creating a block record (a chain of blocks), and finally implementing a consensus protocol.

Thus, validation, verification, consensus, and immutable recording lead to trust and security of the blockchain.

Several types of consensus are used in the blockchain including PoW [21], PoS [21], PoA [21], PBFT [21], Ripple [20], and DAC [21]. In this paper, we will describe only the PoW algorithm that will be implemented in HW (FPGA platform). In the next part, we will describe the state-of-the-art of embedded systems.

3. Overview of Embedded Architectures' Solutions

The evolution of electronics and microelectronics has made it possible to minimize the size of transistors to increase the number of electronic components integrated on the same chip. The main component is the microprocessor. Microprocessors consist of one or more central processing units (CPUs), as well as other modules required for their operation such as memory controllers, cache memory, and I/O controllers.

However, in some systems, the integrated circuit contains not only the microprocessor but also other components such as microcontrollers and GPUs. Such a system is called System on Chip (SoC). These SoCs are based on the minimization of space and power consumption, while preserving the necessary performance for the constraints of the appropriate applications.

For example, a typical modern SoC contains the CPU, the GPU, the communication modules (Wi-Fi, Bluetooth, etc.), a module for localization, as well as other subsystems and coprocessors providing various functions such as device security [9].

These SoCs are used in applied computer systems generally called embedded systems. Although there is no formal definition of the latter, they are generally information systems designed for well-defined tasks [22] and are integrated in other products [23].

The use of embedded systems has also touched the blockchain technology. Thus, e-health, agriculture, light and heavy industry, e-learning, and augmented reality [24] applications are often based on SoCs to set up systems that meet their different needs.

Thus, we find different architectures that are in adequacy with the different needs. We can find single processor systems whose performance is enhanced by HW accelerators (IPs) [24], or massively parallel architectures that take advantage of the large number of processors operating in perfect parallelism [25].

If the use of embedded systems has touched several domains, its use in the blockchain domain has remained rather limited, especially for FPGAs' technology. In fact, despite its various internal resources such as embedded high-speed memory, parallel computing blocks, and flexible architecture, which are suitable for computationally complex applications, it is still limited to the use of the PoW consensus.

Such idea is rarely discussed in the literature. We mention particularly in the work presented in [18], where the authors presented the possibility of implementing an embedded robotics application managed by blockchain.

In the work by Chaari [26], an embedded system based on a Raspberry Pi 3 platform was used. One of the problems encountered in this work is essentially that the Raspberry is unable to run all the PoW consensus software functions due to its limited capabilities.

In this paper, the main target is to propose an embedded architecture suitable for blockchain applications and able to support the implementation of the PoW consensus. Hence, we will show the feasibility as well as the gain realized by using such architecture adopted at Ethereum PoW on FPGAs.

3.1. Ethereum Blockchain Components. In this section, we are interested in blockchain components, especially Ethereum blockchain and its different components.

The blockchain is based on specific terminology representing important concepts. Among the frameworks of the blockchain, there are the following.

3.1.1. Transactions. These are the exchanges of data between different users. Each transaction is signed by the sender's private key. Thanks to this signature, the security of the transactions is guaranteed. Therefore, any modification of these transactions during transmission can be avoided.

3.1.2. Blocks. A block is a record in the blockchain which contains the confirmed transactions. Thus, each open transaction will be added to a block. After a period, for a new block containing transactions to be added to the blockchain, it must be validated by a selected person called a minor. This validation operation is called mining.

3.1.3. The Block Chains. Each block in the blockchain is linked to the previous block. This link is done by inserting the hash specific to the previous block. Therefore, the hash of each block includes not only its own hash but also the hash of the previous block. Figure 1 illustrates what has been described. This way we can protect the blockchain from any form of corruption.

3.1.4. Smart Contracts. A smart contract is a software "installed" on a blockchain solution. It is the most important link in the blockchain. It runs automatically as soon as the various preprogrammed constraints are checked. Even though it is not a legal document, the intelligent contract automates the execution of a contractual commitment.

A consensus algorithm is a process through which all the nodes of the blockchain network achieve a common agreement about the actual state of the distributed ledger [26]. A well-designed consensus protocol can ensure the fault tolerance, authenticity, and security of a blockchain system.

3.1.5. Ethereum Consensus Algorithm. The Ethereum consensus is based on the Ethash algorithm, also known as the Dagger Hashimoto algorithm. The simplified diagram [28] described in Figure 2 represents this algorithm structure and particularly the main one [29].

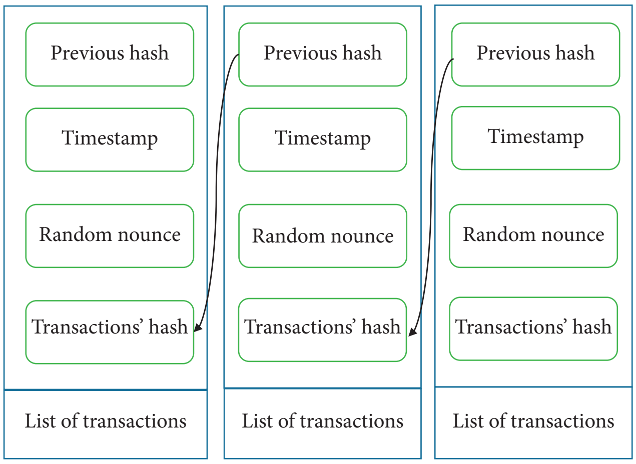


Figure 1: Blockchain illustration.

The profiling of the Ethash algorithm shows that the most used and consuming part is the Keccak 256 part. Therefore, we will implement this part in HW.

4. From SW to HW Architecture

We notice that the implementation of new technologies (IoT, identification, recognition, virtual reality, etc.) is no longer carried out on traditional platforms (PCs, servers, GPUs, etc.) but on embedded systems that can be either generic or well-tailored to the specific requirements of these emerging applications.

To set up a customized solution, it is important to use a mixed SW/HW design allowing adequate mixture of programmability and computing power.

Unlike the development of computer-based software and systems, which is very resource-intensive, the implementation of a System-on-Chip is based on a specific methodology to meet the limitations imposed by the target platforms. In this section, we will characterize the methodology used to realize the design flow of system-on-chip.

The development can be carried out according to several models. The V model presents the development cycle of a system.

This approach is based on two axes:

An axis of specification and design: this axis has as a parameter realization time

An axis of realization and integration: its parameters are the systems and components

Starting from a defined need, the first stage, which is the specification stage, consists of defining the system to be generically realized and then specifying the performances to be respected. Then, the design stage must be implemented. As for the specification, the design is based on two parts: a first generic followed by a second one which is detailed and during which the system is subdivided into different blocks. This conceptual approach leaves room for the realization of the components of our system.

Once the system realization part is completed, a battery of tests is necessary before obtaining our finished product. We start

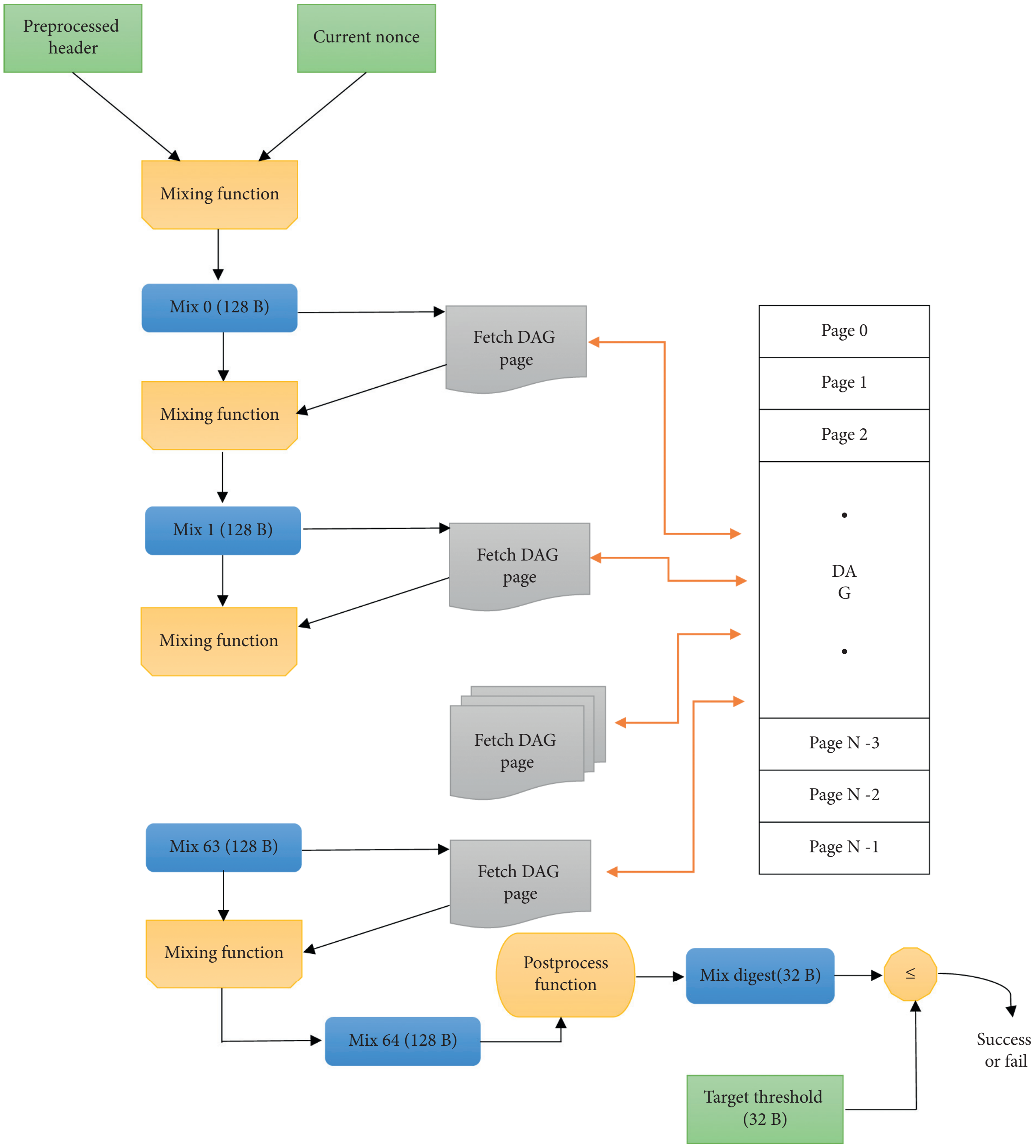


Figure 2: Flow diagram of the Ethash algorithm used by Ethereum with a DAG size of 2.37 GB of late 2018 [27].

with unit tests to verify the functioning of the previously defined blocks. Then, an experiment of integration of these different blocks is carried out. After that, a performance verification is set up to meet the specification presented in the first part. Then, the system integration is done for validation. Finally, an operational test is carried out to verify compliance with the expected specification. This being completed, our product is finalized. It thus meets the need defined previously [21, 30].

5. Embedded System Fields of Application

The use of embedded systems emerges in several fields such as agriculture, industry 4.0, smart cities, and e-health. To design efficient embedded architectures for blockchain applications, we need to profile the consensus algorithm to design an architecture on the FPGA platform. It is possible to have as a result a monoprocessor or a multiprocessor architecture. Different tasks are subdivided on processors during program execution.

In other systems, it is possible to have a monoprocessor architecture with coprocessors (also named IPs). These coprocessors are designed using a HW language such as VHDL, Verilog, System Verilog, and System C.

Such an approach was used for example in the study by Frikha et al. [31], where the authors implemented an adaptive multimedia multiconstraints' system based on dynamic reconfiguration on FPGAs with augmented reality as a case study. In the work by Boutekkouk [32], the author presented the design of an intelligent embedded system. This system can be used in many artificial intelligence-based systems such as expert systems, neural networks, and other sophisticated artificial intelligence (AI) models to guarantee some important characteristics such as self-learning, self-optimising, and self-adaptation.

Among the embedded systems' application fields, we can also mention smart cities [33], smart agriculture [34], and e-health [35]. All these fields based on IoT use embedded systems mainly for their adaptability in designing systems with low energy consumption.

In this paper, we choose a monoprocessor system coupled with hardware accelerators that executes the most complex part of the application. Using the same approach proposed in the study by Frikha et al. [31], we profiled the consensus algorithm proposed by Ethereum. Thanks to this profiling, we will implement the best architecture to minimize the resources and improve the SW execution time.

This will allow us to choose the best possible architecture. We propose to implement an embedded architecture for the Ethereum hash algorithm. This algorithm named Ethash is a SHA 3. The implemented part is the Keccak 256 algorithm.

To the best of our knowledge, this blockchained approach has not been previously implemented. Additionally, the key idea of the work is to address the problem of important energy consumption of public blockchains.

6. The Proposed Consensus Embedded Architecture

Since the PoW consensus algorithm is the most time-consuming and energy-intensive part of the blockchain process, the aim of this paper is to reduce its execution time.

This proposed approach is based on a mixed on-chain and off-chain implementation. Only one part of the implementation (on-chain part) is connected to the blockchain. The other part (off-chain part) is connected directly to the on-chain part, and it is responsible for giving the consensus result.

More precisely, the PoW consensus and, more specifically, the part of the Keccak 256 algorithm on FPGA will do the off-chain encryption.

Inspired by Baklouti and Abid [25], we have set up this system to implement the PoW consensus and more specifically the part of the Keccak 256 algorithm on FPGA to do the off-chain encryption.

Keccak 256 is a part of Ethash which is the consensus of the PoW repetition.

Figure 3 represents the Keccak deployment architecture.

In this section, we are going to compare the software implementation and the hardware implementation of the Keccak hash algorithm. After profiling, Keccak is the more complex, energy consuming, time consuming, and repeated function.

As input of the Keccak system, we have the proposed new block, the head of the most recent block, and finally the nonce value. The hash and the combination of different blocks give a hash number. If this number is less than the target value, then we solved the PoW, else we must increment with a new nonce value and try the whole process again.

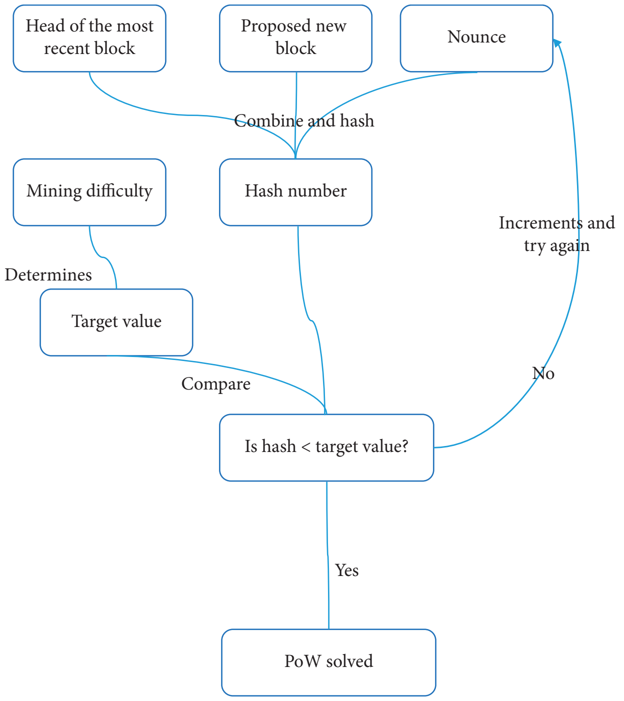


Figure 3: Keccak implementation algorithm.

The mining difficulty was determined by comparing the hash number and the target value. As mentioned in the work by Chaari [26], the implementation of the blockchain Ethereum node on a resource-constrained platform such as the Raspberry PI3 shows that the implementation of PoW leads to the platform crash.

As a first contribution, we present here the study we carried out in order to divide our node on two parts: a node without PoW that works on-chain: it runs on the ARM processors of the Raspberry Pi 3, and an off-chain verification part implemented on FPGAs.

In the following section, we will describe the obtained results and the implemented system.

7. Experimental Results

7.1. Initial System. After writing our genesis file and running the init command on the Raspberry Pi 3, the initialization of our blockchain was successful. Then, we were able to execute the node and access the JavaScript console where we performed some basic ether transfer transaction between the predefined accounts which were successfully submitted. But the moment the mining is being started, the Raspberry Pi 3 would overheat and stopped functioning. For that, we executed another node from the same blockchain on a computer that was able to mine the transactions and synchronize the results with the node running on the Raspberry Pi 3 as illustrated in Figure 4. Therefore, using Proof of Work, a Raspberry Pi3 can only synchronize the mined blocks but not mine new ones. That is why we decided to implement consensus system off-chain.

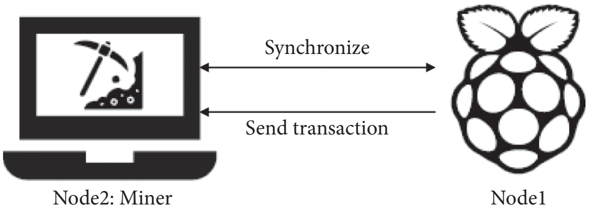
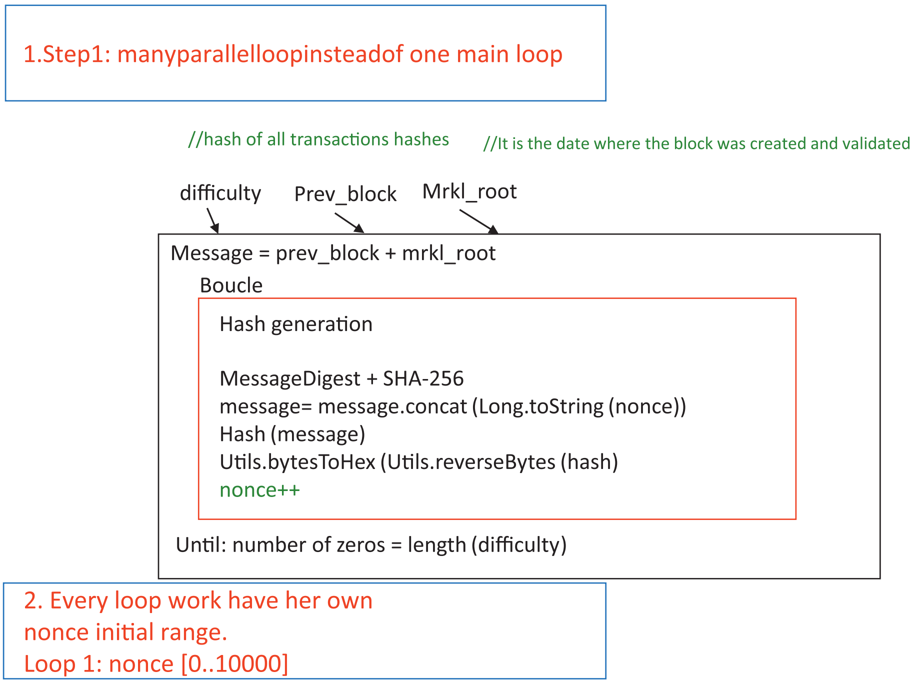


Figure 4: Private Ethereum blockchain using PoW consensus.



1.Step1: manyparallelloopinsteadof one main loop

//hash of all transactions hashes

//It is the date where the block was created and validated

difficulty Prev_block Mrkl_root

```
Message = prev_block + mrkl_root
   Boucle
      Hash generation

      MessageDigest + SHA-256
      message= message.concat (Long.toString (nonce))
      Hash (message)
      Utils.bytesToHex (Utils.reverseBytes (hash)
      nonce++

 Until: number of zeros = length (difficulty)
```

2. Every loop work have her own nonce initial range.
Loop 1: nonce [0..10000]

Figure 5: SW profiling result.

7.2. Keccak FPGA Implementation

7.2.1. Code Profiling Result. By taking the code implemented in the Java language related to the Ethereum node, we managed to isolate the part corresponding to the PoW consensus. This code has also been profiled to obtain the result of Figure 3. The result of this profiling is described in Figure 5.

Several loops are present: the relative loop to the nonce is repetitive and independent of any other input. We can consequently implement any VHDL system and create several generators of nonce values.

7.2.2. VHDL Keccak Implementation. Due to the health crisis and the impossibility to have more performant platforms, we choose to use the available ones. Henceforth, we use the Raspberry Pi 3. For the ZedBoard, we can explain it to its outperformance compared to the Virtex 5 ML 507 one. The implementation of the Keccak code in VHDL has been done to create an ASIC allowing the working off-chain to do the hash and to set up the PoW consensus. We used the Xilinx ZedBoard FPGA as a prototyping platform to realize the Keccak [29]. This board is an evaluation and a development board based on the Xilinx Zynq 7000.

Combining a dual Cortex-A9 Processing System (PS) with 85,000 Series-7 Programmable Logic (PL) cells, the Zynq-7000 AP SoC can be targeted for broad use in many applications. The ZedBoard's robust mix of on-board peripherals and expansion capabilities make it an ideal platform for both novice and experienced designers [29].

To improve this system, we have added 4 independent IPs to generate the nonce values. As an example, in [0.10000] interval, we are able to allocate to the IPs 1, 2, 3, and 4, respectively, the intervals [0.249], [250.499], [500.749], and [750.1000].

Figure 6 represents the proposed architecture of Keccak RTL implementation architecture. It contains different inputs and outputs but also the logic gates, Fifo, Padder bloc, Hash bloc, and different RTL registers.

7.3. Simulation Results and Comparison

7.3.1. Simulation Results. After the simulation results of the Keccak RTL implementation, the VHDL code simulation is proposed in Figure 7. The value of nonce to obtain the hash value is indicated in the figure by the arrow. Note that the nonce value used to obtain the proper hash is 239327.

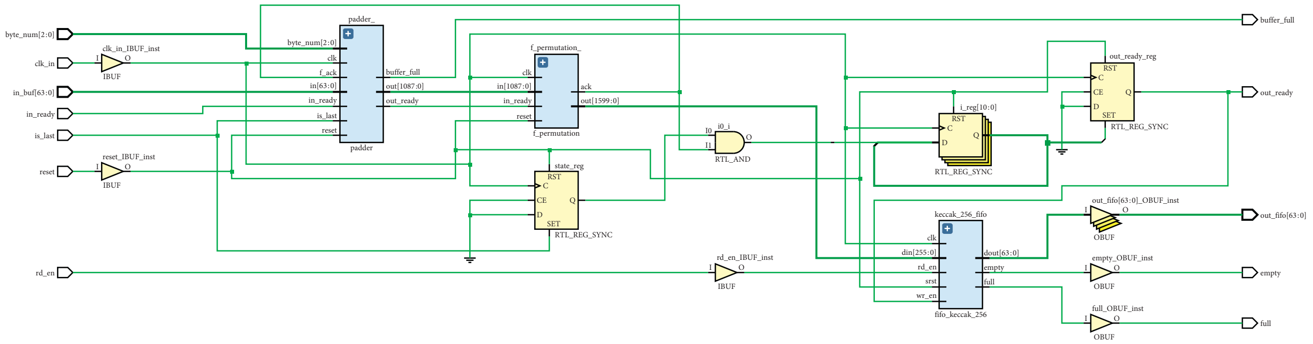
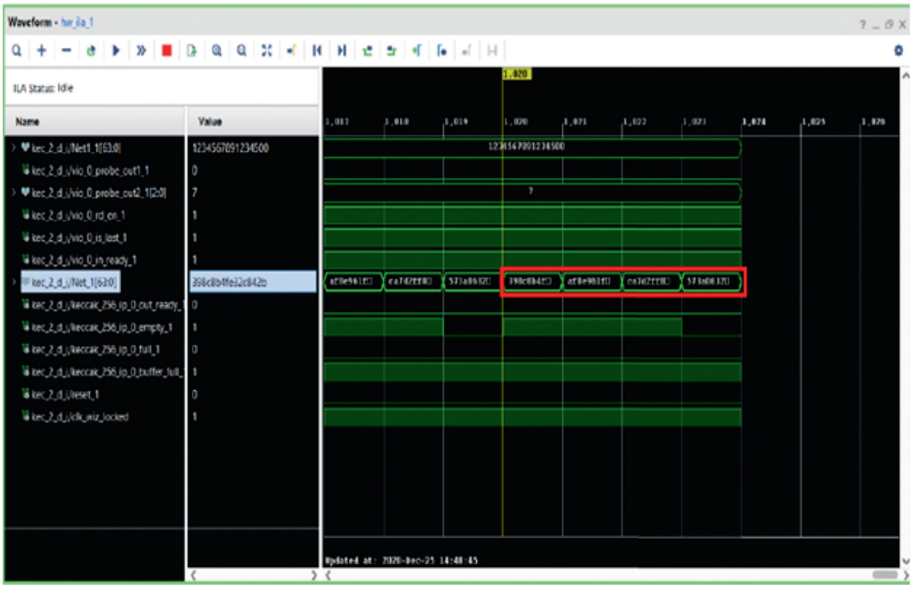


Figure 6: KECCAK RTL implementation.



Waveform - hw_ila_1

ILA Status: Idle

Name	Value
kec_2_d_i/Net1_1[63:0]	1234567891234500
kec_2_d_i/vio_0_probe_out1_1	0
kec_2_d_i/vio_0_probe_out2_1[2:0]	7
kec_2_d_i/vio_0_rd_en_1	1
kec_2_d_i/vio_0_is_last_1	1
kec_2_d_i/vio_0_in_ready_1	1
kec_2_d_i/Net_1[63:0]	398c8b4fe32c842b
kec_2_d_i/keccak_256_ip_0_out_ready_1	0
kec_2_d_i/keccak_256_ip_0_empty_1	1
kec_2_d_i/keccak_256_ip_0_full_1	0
kec_2_d_i/keccak_256_ip_0_buffer_full_1	1
kec_2_d_i/reset_1	0
kec_2_d_i/clk_wiz_locked	1

Updated at: 2020-Dec-25 14:48:45

Figure 7: KECCAK simulation result.

Table 1: HW/SW comparison.

	SW	HW1	HW2
Execution time (ms)	21	3.98	2.78
Energy consumption (W)	3.7	1.2	1.7

7.3.2. SW and HW Comparison. After implementing the code, we tried to compare the SW version of the code implemented in Java running on Raspberry PI 3 and the two architectures. The HW1 architecture represents the complete implementation on the Keccak code presented in Figure 3. HW2 consists of using 4 nonce-generating IPs working in parallel in order to parallelize the code and to minimize the execution time.

We notice that the HW1 gain compared to the SW is approximately 5.25x. The HW2 gain compared to the SW is approximately 7.55x. The energy consumption on the Raspberry PI 3 is 3.7 W; however, in the HW version, we note that the HW1 requires 1.2 W, while the second requires 1.7 W.

The difference in consumption despite the use of the same platform (ZedBoard) for the HW1 and HW2 is due to the duplication of the IPs of nonce generators.

Table 1 illustrates the obtained result of HW/SW comparison.

The system obtained after this implementation is described in the figure. We can find there a description of the classical architecture of Ethereum followed by the on-chain/off-chain architecture that has been adopted.

Figure 8 presents the proposed part in the paper with an on-chain architecture implemented on Raspberry Pi3, whereas the offchain one is set up on FPGA.

Figure 9 represents a comparison between the classical blockchain architecture and the proposed architecture.

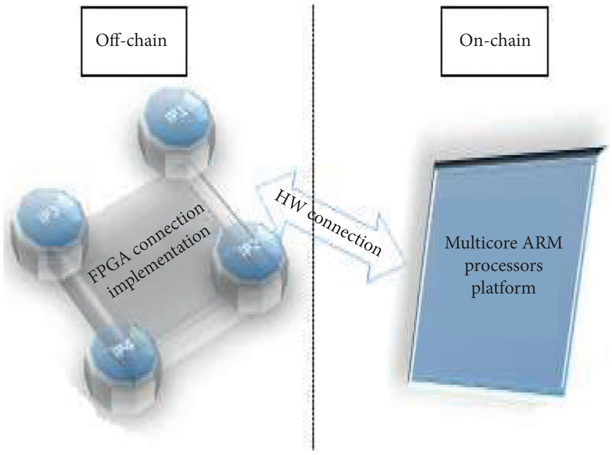

Off-chain
On-chain
FPGA connection implementation
HW connection
Multicore ARM processors platform

Figure 8: Mixed architecture for PoW algorithm implementation.

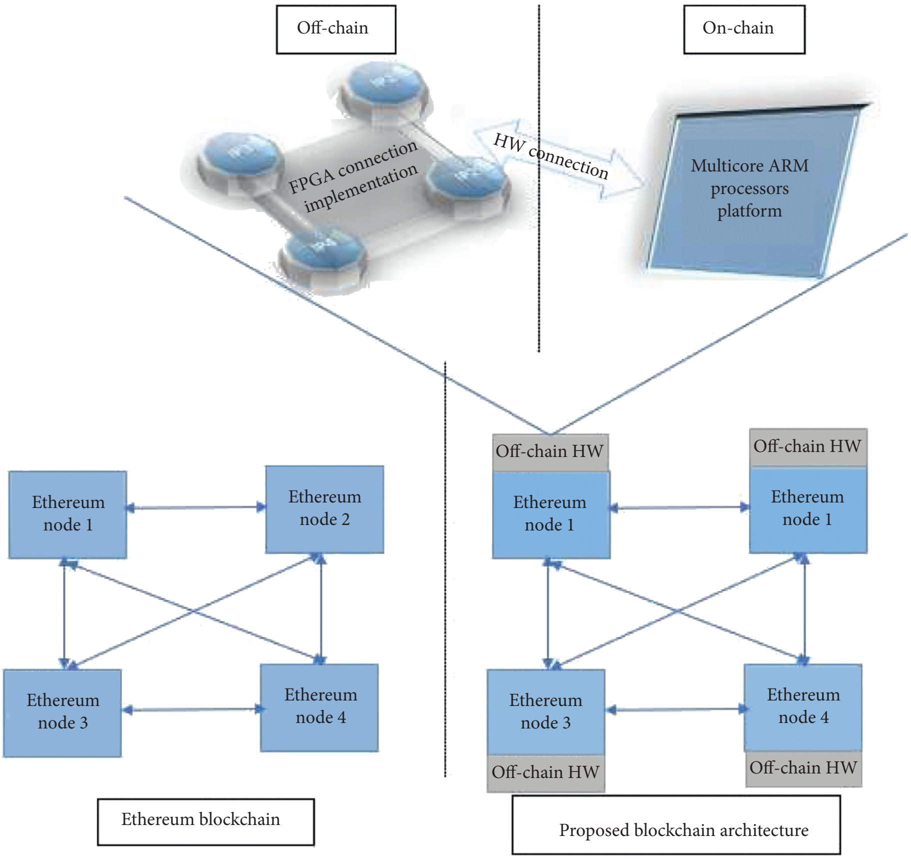


Figure 9: Whole proposed architecture implementation.

8. Conclusion

In this paper, we have highlighted the HW implementation of the PoW consensus. This consensus is used in the Ethereum blockchain. We were able to demonstrate that, to successfully implement this consensus on low-resource platforms, it is possible to use an on-chain system to successfully transfer and receive data and an off-chain system to implement the consensus and send the result to the on-chain node. This system, despite its complexity, allows a gain of at least 5 times compared to a pure SW system in execution time, while minimizing energy consumption. It can also be improved and accelerated by playing on the different blocks of the consensus. Indeed, we have added 4 IPs of nonce generators, but we could improve the result even more by adding more Keccak 256 and or 512 IPs to have a more efficient and faster system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Dr. Omar Cheikhrouhou thanks Taif University for its support under the project Taif University Researchers Supporting Project (no. TURSP-2020/55), Taif University, Taif, Saudi Arabia.

References

[1] H. Treiblmaier, A. Rejeb, and A. Strebinger, "Blockchain as a driver for smart city development: application fields and a comprehensive research agenda," *Smart Cities*, vol. 3, no. 3, pp. 853–872, 2020.

[2] G. Bloom, B. Alsulami, E. Nwafor, and I. C. Bertolotti, "Design patterns for the industrial internet of things," in *Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pp. 1–10, Imperia, Italy, 2018.

[3] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," *SSRN Electronic Journal*, 2008.

[4] A. Polyviou, P. Velanas, and J. Soldatos, "Blockchain technology: financial sector applications beyond cryptocurrencies," *Proceedings*, vol. 28, no. 1, p. 7, 2019.

[5] C. Agbo, Q. Mahmoud, and J. Eklund, "Blockchain technology in healthcare: a systematic review," *Healthcare*, vol. 7, no. 2, p. 56, 2019.

[6] V. Paliwal, S. Chandra, and S. Sharma, "Lockchain technology for sustainable supply chain management: a systematic literature review and a classification framework," *Sustainability*, vol. 12, 2020.

[7] J. Lee, M. Azamfar, and J. Singh, "A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems," *Manufacturing Letters*, vol. 20, pp. 34–39, 2019.

[8] G. Pau, M. Collotta, A. Ruano, and J. Qin, "Smart home energy management," *Energies*, vol. 10, no. 3, pp. 382–386, 2017.

[9] G. Mirabelli and V. Solina, "Blockchain and agricultural supply chains traceability: research trends and future challenges," *Procedia Manufacturing*, vol. 42, pp. 414–421, 2020.

[10] K. Cho and Y. Cho, "HyperLedger fabric-based proactive defense against inside attackers in the WSN with trust mechanism," *Electronics*, vol. 9, 2020.

[11] N. Wang, X. Zhou, X. Lu et al., "When energy trading meets blockchain in electrical power system: the state of the art," *Applied Sciences*, vol. 9, 2019.

[12] D. Siswantoro, R. Handika, and A. F. Mita, "The requirements of cryptocurrency for money, an Islamic view," *Heliyon*, vol. 6, no. 1, 2020.

[13] B. Tavares and F. Figueiredo Correia, "A survey on blockchain technologies and research," *Journal of Information Assurance and Security*, vol. 14, pp. 118–128, 2019.

[14] K. Christodoulou, E. Iosif, A. Inglezakis, and M. Themistocleous, "Consensus crash testing: exploring ripple's decentralization degree in adversarial environments," *Future Internet*, vol. 12, 2020.

[15] M. Salimitari, M. Chatterjee, and Y. P. Fallah, "A survey on consensus methods in blockchain for resource-constrained IoT networks," *Internet of Things*, vol. 11, 2020.

[16] https://www.thebalance.com/how-much-power-does-the-bitcoin-network-use-391280#:%7E:text=Regardless%20of%20the%20number%20of,usage%20provided%20by%20ASIC%20miners.

[17] https://zipmex.com/learn/how-long-to-mine-ethereum/#:%7E:text=Successful%20mining%20on%20the%20Ethereum,a%20block%20of%20Bitcoin%20transaction.

[18] Y. Sakakibara, Y. Tokusashi, and H. Matsutani, "Accelerating blockchain transfer system using FPGA-based NIC," in *Proceedings of the 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications*, pp. 171–178, Melbourne, Australia, December 2018.

[19] Z. Tu and C. Xue, "Effect of bifurcation on the interaction between Bitcoin and Litecoin," *Finance Research Letters*, vol. 31, 2019.

[20] S. Bamakan, A. Motavali, and A. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, 2020.

[21] T. Frikha, H. Choura, N. Abdennour, O. Ghorbel, and M. Abid, "ESP2: embedded smart parking prototype," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 6, pp. 1569–1576, 2020.

[22] P. Marwedel, *Embedded System Design: Embedded Systems, Foundations of Cyber-Physical*, Springer International Publishing, Berlin, Germany, 2018.

[23] T. Noergaard, "Embedded systems architecture: a comprehensive guide for engineers and programmers," in *Embedded Systems Architecture*, pp. 261–293, Elsevier Science, Amsterdam, Netherlands, 2013.

[24] T. Frikha, N. Ben Amor, J.-P. Diguet, and M. Abid, "A novel Xilinx-based architecture for 3D-graphics," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14947–14970, 2019.

[25] M. Baklouti and M. Abid, "Multi-softcore architecture on FPGA," *International Journal of Reconfigurable Computing*, vol. 2014, Article ID 979327, 13 pages, 2014.

[26] A. Chaari, "*Storing health and fitness data on an ethereum based blockchain*," M.S. thesis, National Engineering School of Sfax, Sfax, Tunisia, 2019.

[27] Unknown, ETHASH., 2018, https://miningbitcoinguide.com/mining/sposoby/ethash.

[28] M. Stachowski, A. Fiebig, and T. Rauber, "Autotuning based on frequency scaling toward energy efficiency of blockchain algorithms on graphics processing units," *The Journal of Supercomputing*, vol. 77, no. 1, pp. 263–291, 2020.

[29] Xilinx, Zeadboard User Guide, 2013..

[30] S. Falcone, J. Zhang, A. Cameron, and A. Abdel-Rahman, "Blockchain design for an embedded system," *Ledger*, vol. 4, no. 1, 2019.

[31] T. Frikha, N. Ben Amor, K. Lahbib, J. P. Diguet, and M. Abid, "A data adaptation approach for a HW/SW mixed architecture (case study: 3D application)," *WSEAS Transactions on Circuits and Systems*, vol. 12, no. 9, pp. 263–272, 2013.

[32] F. Boutekkouk, "Embedded systems codesign under artificial intelligence perspective: a review," *International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 32, no. No. 4, 2019.

[33] A. Kumar and V. Nath, "Study and design of smart embedded system for smart city using internet of things," in *Nanoelectronics, Circuits and Communication Systems*, V. Nath and J. Mandal, Eds., Springer, Singapore, Singapore, 2019.

[34] M. Mahbub, "A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network," *Internet of Things*, vol. 9, 2020.

[35] A. J. Bokolo, "Application of telemedicine and eHealth technology for clinical services in response to COVID-19 pandemic," *Health and Technology*, vol. 11, no. 2, pp. 359–366, 2021.